**The Report Committee for Debra Lynn Ayers**
**Certifies that this is the approved version of the following report:**

# Consumer Preference Measurement and Its Practical Application for Selecting Software Product Features

**APPROVED BY**
**SUPERVISING COMMITTEE:**

**Supervisor:** ______________________________
Kyle Lewis

______________________________
Kathleen Mackie

# Consumer Preference Measurement and Its Practical Application for Selecting Software Product Features

**by**

**Debra Lynn Ayers, B.S.**

**Report**

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

**Master of Science in Engineering**

**The University of Texas at Austin**

**August 2011**

## Acknowledgements

I would like to express gratitude to my management at Oracle for supporting my efforts to research this topic in the context of my role as a Sr. Product Manager for Fusion Middleware Business Integration Division. A special acknowledgement to Kate Mackie, whose sage advice and patient encouragement helped me complete this project. Kate's teachings in Marketing sparked my curiosity and interest to apply advanced preference measurement to aid with complex software product management decisions. Last but not least, a special thank you to my family who have encouraged me to seek the truth in every endeavor.

*We shall never cease from exploration*

*And the end of all our exploring*

*Will be to arrive where we started*

*And know the place for the first time.*

***-T. S. Eliot (1888 - 1965)***

# Abstract

## Consumer Preference Measurement and Its Practical Application for Selecting Software Product Features

Debra Lynn Ayers, M.S.E.

The University of Texas at Austin, 2011

Supervisors: Kyle Lewis, Kathleen Mackie

Consumer preference measurement is a quantitative field of study for modeling, collecting and analyzing product decisions by consumers. Discovering how consumers choose products is an important area of marketing research and recognized as a successful partnership between academic theory and practice over the past forty years. Despite preference measurement's success in consumer products, little guidance is available for its application to software product management. This paper assesses the feasibility of applying advanced preference measurement techniques to software products and suggests a framework for conducting such studies. A summary of the methods is provided to give guidance to software product managers seeking to apply preference measurement to common product decisions. The paper concludes by recommending a technique called '*maximum difference scaling*' to elicit customer feedback to help measure the importance of new features for software product improvement.

# Table of Contents

## List of Tables

# List of Figures

# List of Illustration

# Chapter 1: Introduction

## Motivation

How does a product manager decide what new features should go into a new product or enhance an existing one? In a highly competitive marketplace, where products and pricing are constantly changing, how does one build repeatable, reliable processes for product feature selection, ensuring engineering investments elicit positive customer response to boost sales and profitability? Traditional consumer products companies, with established brand marketing divisions, leverage consumer preference measurement to test and validate new product concepts with potential customers. Consumer products companies leverage advanced marketing tools to mitigate risk. Moreover, these tools provide assurance that limited engineering and manufacturing resources are spent building the right products and sales are targeting the right customers.

Software products companies, a relatively young industry in comparison to consumer goods, have grown rapidly through innovation fueled by the proliferation and affordability of personal and mid-range computers. For example, Oracle Corporation, distinguished as an early pioneer in the software industry, was founded only thirty years ago and is now listed in the top one-hundred most profitable companies by Fortune Magazine (Fortune Magazine 2009). While notable exceptions exist, most software companies have prospered through engineering innovation with limited customer interaction in the product concept and design phases. However, as the Information Technology sector has stabilized and matured over the past decade, software products are now increasingly marginalized and commoditized (Car 2003). The pressure of fierce competition will force software companies to seek and borrow marketing techniques from established consumer product industries to differentiate and thrive in a customerdriven market place as other break-through industries have done in the past, computer hardware and automotive industry as recent examples. To stave off further pressure from commoditization, product strategists should consider employing advanced marketing techniques such as *conjoint*[1] studies, to aid in product design and pricing (Gownder 2011). These studies can inject customer feedback directly into the product concept and development phases to guide decisions and ensure investments are being made where they matter most to customers.

In summary, the era of pure innovation as a marketing strategy is diminishing for the software industry. Traditionally, consumers in mature industries decide what features are essential and what prices they are willing to pay for those features. Software product managers and strategists may routinely bring customers into product design and roadmap discussions via customer visits and focus groups; however, there is a need for more systematic ways to incorporate this feedback into product development decisions.

### STATEMENT OF PURPOSE

Consumer preference measurement has evolved over the past forty years and is widely lauded as one of the most successful cross-over breakthroughs between academic theory and practice in the field of marketing research. (Green, Krieger, and Wind 2001) (Bradlow 2005). The techniques have been successfully applied by hundreds of companies helping make decisions in critical areas of product design, branding, segmentation, targeting and price feature comparisons by revealing what consumers like and prefer (Dolan 1990) (Gownder 2011). While the literature reveals great success applying preference measurement in traditional consumer goods (Cattin and Witnik 1989), its use is not yet well-established for software product strategy and management.

[1] Conjoint analysis is also known as trade-off analysis deriving from the word conjoint which means to consisting of, or involving two or more combined or associated entities.

This paper assesses the feasibility of applying advanced preference measurement techniques to software products and suggests a framework for conducting such studies. The remainder of the paper is structured as follows: Chapter 2 reviews the most relevant preference measurement designs and details how conjoint methods work. These are presented with a historical perspective to give an appreciation for the evolution of the field. Chapters 3 presents an empirical study to apply *maximum difference scaling* to help select software product features for product line extension. Chapter 4 closes the paper with lessons learned and recommendations for future investigations to improve the application of consumer preference measurement for complex software feature sets. A summary of recommended methods is included to guide software product managers in the use of consumer preference measurement to guide product decisions.

# Chapter 2: Methodological Background

## PREFERENCE DESIGN AND DATA COLLECTION

Preference measurement grew from work in mathematical psychology in the late 1960's. Quite recently this work has been traced to the 1920's, with the research by L.L. Thurstone on the laws of *Methods of Paired Comparison* (Thurstone 1927). This body of original work centered on interviewing consumers and devising ways to measure their preferences for goods and services through quantitative analysis (Johnson 2005).

Practitioners observed that when consumers were asked to judge product features in isolation, the results were inconsistent and unpredictable compared to when those same features were presented in a different context. For instance, fragrance may be an important selection criterion for buying fabric softener; however, if all fabric softeners smell exactly the same, fragrance becomes unimportant,[2] and other product features are the basis for a product decision. This observation led to the premise that consumers' behavior is governed by *trade-offs*. Although a consumer may not be able to articulate an overall judgment if asked directly about a single attribute, when forced to make difficult trade-offs and concessions, the true value of the product features are revealed by the consumer (Green and Rao 1971) (Johnson 1974). Hence, trade-off or conjoint analysis ("CA") was born. The strength of conjoint is its ability to ask realistic questions that mimic the trade-offs that consumers make in the real world. The next sections will walk through the most prevalent conjoint methods, highlighting strengths and weaknesses.

### Traditional Conjoint Analysis

The fundamental idea in conjoint is that a product can be decomposed into a set of relevant attributes or features (Green and Rao 1971). For example, one might describe a

[2] Fragrance may be a commodity feature of the product.

software product by attributes such as brand, messaging protocol and caching, as illustrated in Table 1.

| Brand | **Messaging Protocol** | **Caching** |
|---|---|---|
| **Oracle** | **JMS** | **Yes** |
| **IBM** | **MQ Series** | **No** |
| **TIBCO** | **RV** | **No** |

**Table 1**: Attributes with Levels for Software Product

The *levels* of an attribute are the possible values an attribute may have. For example in Table 1 above, brand is the attribute and the possible levels are Oracle, IBM, TIBCO in column one. By varying the levels of the attributes across a set of product profiles and asking an individual to give an overall judgment of preference, a value system can be derived across the set of profiles. An example of a product profile is shown in Illustration 1.

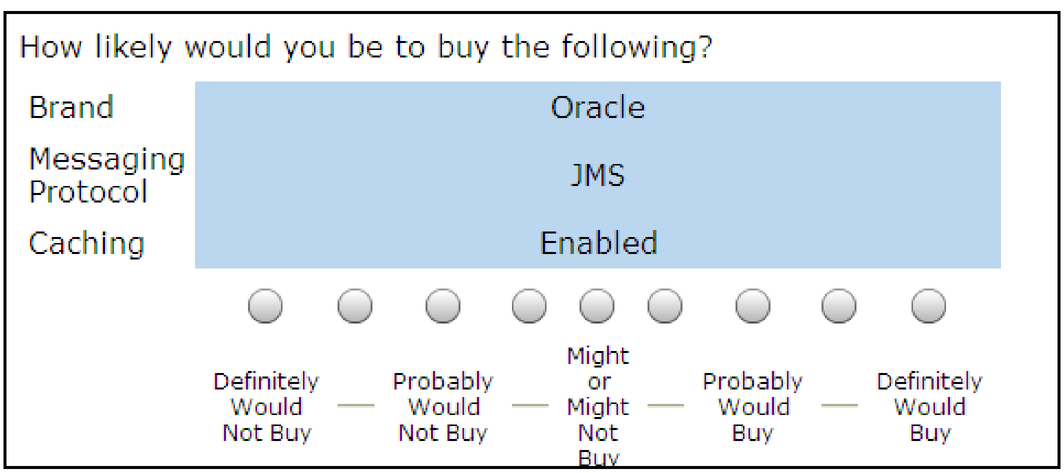



**Illustration 1**: Traditional CA Product Profile

Once a value system is established by asking the consumer to rate every possible product profile, insight can be gleaned as to what attributes the individual is willing to trade-off for others. One can also predict the product most likely to appeal to that individual by comparing new product profiles to their established value system.

The value system is constructed in the following way. After preferences are gathered from an interview asking the consumer to rank or rate profiles, mathematical analysis is performed on the collected preference data. The product profiles, represented as attribute contributions, are encoded into a trade-off matrix. Each combination of attributes represents a vector in the matrix. Using various linear estimation algorithms, a weight or *partworth* is calculated for each of the attribute levels. Once a partworth has been determined for all levels, the respondent's overall *utility* or preference for a given product can be estimated by summing the partworths for each level of attribute that describes that product. These utilities define a value system unique to that individual. Refer to Appendix A, Figure 1 for an example of partworths, utilities and attribute importance calculations. Performing analysis on many individuals can allow one to segment customers to understand which attributes or features are most influential in the product selection decision process for a given study.

There are limitations to CA, in general, and in *traditional* CA, in particular. It is important to understand these limitations to gain an appreciation for how the field has evolved over time. It is essential for applying conjoint to more complex scenarios. First, CA is based on an additive model to derive part worth estimates. This model requires the attributes selected for study be *independent* and *non-redundant* (Johnson 1974). The model assumes, for instance, the extent to which a customer prefers blue packaging to red packaging is independent of size, weight or price. In the example in Table 1 above, this principle says that the brand preferred should be independent of whether caching is preferred by the respondent.

Product attributes must be independent. Attribute independence is imposed by the linear principles of the mathematical models typically used to estimate preference data, namely multiple linear regressions, logistic regression or *logit model*s [3] (Green and Rao 1971). If variables are not independent, the results produced from the model will be *biased*– the weights assigned will be skewed because linear equations cannot represent these hidden co-dependencies. In conjoint literature, these are called *main effects* and *interaction effects* respectively. Main effects ignore the possibility of interactions between attributes: designs that follow these principles are said to be *orthogonal*. Orthogonal designs assure than an estimate of one attribute in unaffected by the estimate of other attributes. In practice, this principle of orthogonality plays an important role in increasing the robustness of conjoint measurements, making it more unlikely to produce counter-intuitive results. This robustness is said to contribute to the managerial satisfaction of conjoint study results (Huber 2005). If interactions of attributes do exist, they must be accounted for in the design of the study (Orme 2010).

There is a practical impact of these rules: the person constructing the conjoint model must have a thorough understanding of the product feature set from a customer's perspective to ensure they are selecting product attributes that are truly independent as selection criteria. If the attributes selected for the study are independent or at least interactions accounted for in the design, the estimation algorithms will produce sound results.

---

[3] In statistics, logistic regression (sometimes called the logistic model or logit model) is used for prediction of the probability of occurrence of an event by fitting data to a logit function logistic curve.

Traditional CA, as introduced, (Green and Rao 1971) (Johnson 1974), works well for small sets of attributes and levels, proven in hundreds of documented trials and studies by researchers and practitioners in the 70's (Cattin and Witnik 1982). However, since traditional CA required a respondent to evaluate all possible combinations of the attributes, only a small set of attributes can be realistically judged before exhausting the respondent. This greatly limits its application for more complex product decisions tasks as only the simplest problems can be modeled effectively. For example, if one designed a study with 3 attributes and 2-3 levels for each attribute, similar to the model depicted in Table 1 above, the respondent would need to evaluate 18 product profiles for the model to be valid.

**3 Brands x 3 Messaging Protocols x 2 Caching = 18 profiles**

The number of profiles to be judged grows factorial by number of attributes and levels; hence it is termed a *full-factorial* design. Illustrating this point further, a full factorial study administered as preliminary research consisting of only 3 attributes for a software product, 18 profiles overall, had dismal feedback from respondents (Ayers 2006). Even though the task to rate each software product profile by preference was simple in concept, complaints were received from respondents regarding the tediousness of the task. Only fifty percent of the respondents returned the surveys completed, some simply rating the profiles for which they had a strong preference. [4] The necessity to evaluate every profile proved so severe for conjoint practitioners in the late 1970's, researchers were inspired to improve the technique and evolved a new method called *adaptive* conjoint analysis.

---

[4] Refer to Appendix A, Figures 1-4 for highlights of a traditional CA study for software product feature selection. This can help one understand why traditional CA is not suitable for complex models such as software product design.

| Traditional CA Strengths | Traditional CA Weaknesses |
|---|---|
| • Shows products in Full Profile, mimicking real world decisions<br>• Can be used for product design and pricing decisions<br>• Captures individual level preferences for all product alternatives. | • Small number of attributes (3-4 due to respondent fatigue)<br>• Limited ability to measure cross-effects of attributes (price vs. brand for example) |

**Table 2**: Traditional CA-Strengths and Weaknesses

**Adaptive Conjoint Analysis**

With the recognition that the current techniques could not scale effectively past a relatively small number of attributes and levels,[5] an improved technique was developed in 1985 with the aid of the personal computer (Johnson 2005). A new estimation technique was employed to calculate attribute utilities; subsequently, the task of interviewing respondents could be simplified. By asking respondents to first rate the relative importance of attributes and levels, they then only need to evaluate a subset of product profiles containing the attributes which have most relevance – usually 2 to 5 – for any one question. These are called *partial* profiles. The term *adaptive* refers to the interview being conducted by computer and being customized for each respondent based on the answers to the importance of attribute questions in the beginning. At each step, previous answers are used to decide which question to ask next, to obtain the most information about the respondent's preferences.

[5] While the literature claim traditional CA can be leveraged for up to six attributes (Green and Srinivasan 1978), research for this paper indicated an unacceptable level of respondent fatigue for only 3 attributes with 3-4 levels each. This could be in part to the complexity of the information presented, as is common in software product design tasks.

With these improvements in place, Adaptive Conjoint Analysis ("ACA") became the most widely used conjoint method in the late 1980's and was lauded by the seminal Harvard Business Review article, *Conjoint Analysis for Managers* (Dolan 1990) bringing attention of the success of conjoint studies to a new, wide audience of business and consumer product managers with claims of easy to use software on personal computers.

| ACA Strengths | ACA Weaknesses |
|---|---|
| • Ability to measure many attributes (30+) without wearing out respondent<br>• High ratio of information gained per respondent | • Partial profiles are less realistic<br>• Not suitable for price research since price is not always included as attribute<br>• Can only be administered via computer (PC or Web) |

**Table 3**: Adaptive CA-Strengths and Weaknesses

**Choice-Based Conjoint Analysis**

Choice-Based Conjoint ("CBC"), also known in literature as Discrete Choice models[6], has theoretical relationship to work done by Thurstone in the 1920's (Thurstone 1927), Luce in 1950's, and McFadden in 1970's.[7] Market researchers started leveraging and expanding upon the technique in the early 1980s (Louviere and Woodworth 1983).

Recall that in traditional CA and ACA studies, respondents are asked to rank or rate product profiles and then utilities are then estimated for <u>each</u> attribute and level across a set of profiles, full or partial. The benefit of this approach is that information is gathered at the individual level. In contrast, CBC allows the respondent to directly choose the preferred product from set of products profiles or concepts, rather than by rating or ranking them individually, as shown in Illustration 2.

[6] Discrete choice models statistically estimate the probability that a person chooses a particular alternative.
[7] Daniel McFadden won the Nobel Prize in 2000 for his pioneering work in developing the theoretical basis for discrete choice.

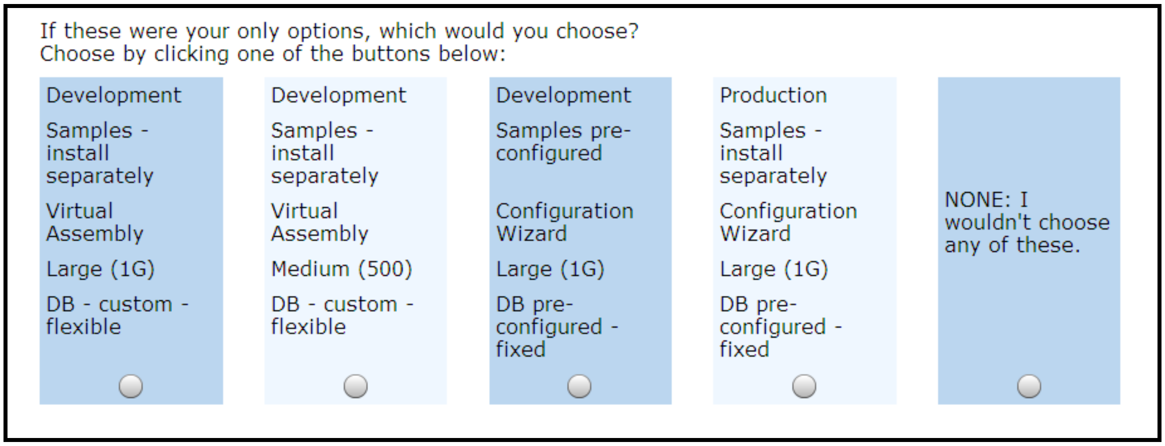



**Illustration 2**: Choice-based Interview for Software Feature Selection

CBC is favored by researchers because the task of choosing a preferred concept is similar to what buyers actually do in the marketplace. Selecting a preferred product from a group is a simple and natural task that anyone can understand. Respondents like this style as it is more intuitive than assigning numerical rankings or ratings, even though initially more information must be reviewed. Unfortunately, CBC, when first introduced, had defects so severe from the market researcher's point of view that its adoption was thwarted for almost a full decade (Sawtooth 2008).

What was so wrong with CBC from the researcher's perspective? While CBC provides a more natural interview style, not as much preference information is revealed from the interview. When a choice is made, one only knows *which* product is preferred not the strength of the preference. Estimation techniques available in the 1990's, namely multinomial logit, aggregated all respondent information together and then generated group-level attribute utilities or importance. Aggregate estimation assumes all respondents have similar interests and selection criteria, i.e. *homogenous*; however, if the respondent pool is quite diverse, i.e. *heterogeneous*, the results are not that meaningful. If one needs to understand the differences in attribute importance across different respondent segments, then traditional CA and ACA were better conjoint alternatives even with their known limitations. Traditional CA and ACA had the advantage of providing individual utilities that could then later be aggregated for group-level analysis. This was not true of CBC in late 1990's and early 2000's. As a result, ACA was the most widely employed conjoint method until the late 1990s when things changed: researchers started leveraging CBC in strong numbers (Sawtooth 2008).

What changed to cause adoption of CBC when it was rejected before? The estimation algorithms were improved. Computer hardware became cheaper and faster so the advanced algorithms could be implemented in software for wide use by market researchers (Orme 2010). Hierarchical Bayes Estimation ("HB"), a statistical distribution algorithm (Rossi and Allenby 2003), allowed individual level data to be estimated from sparse CBC data. Latent Class Algorithms ("LCA") allowed one to simultaneously discover latent, or *ad-hoc* segments within sparse CBC data. Sawtooth Software, the leader in conjoint software, introduced software modules to implement these advancements and they have become widely adopted. Over the past decade, CBC has become the most widely used conjoint-related method, estimated to be at 90 percent of preference measurement studies (Hill and Orme 2011).

How does one design a CBC study? Product concepts are described by attributes and levels as with other conjoint methods. Then, the number of product concepts or profiles to show the respondent at one time, called *tasks*, must be determined. An example of a single task of product profiles is depicted in Illustration 2 above. Finally, one must figure out how many tasks should be presented overall to be a statistically sound study. A balance must be achieved between how many tasks must be shown for the study to be considered valid. The more tasks shown, the more likely the respondent will fatigue and increase risk of collecting meaningless *clickthrough* data. It is common to generate several CBC designs, possibly trimming attributes and levels, thus reducing the number of tasks to achieve good balance. Software for conducting CBC interviews will dynamically generate the product concepts and tasks once the attributes, levels, concepts per task and number of tasks are specified (Hill and Orme 2011). The software will analyze the CBC design for statistical efficiency and warn if not enough tasks are configured for the number of attributes and levels.

Other considerations must also be made such as determining if any combinations do not make sense to show together in the same concept, termed *prohibitions*. CBC designs allow tasks to include a *None* option as depicted in Illustration 2; however, its use is discouraged in much of the literature because no preference information is collected when it is provided. Imagine a respondent selecting all "none" options – no data would be gathered.

As discussed previously, CBC is an inefficient way to elicit preferences at the individual level if aggregate data analysis is leveraged. However, by leveraging the latest methods of preference estimation, namely Hierarchical Bayes estimation and Latent Class, the simplicity of the design far outweighs the early concerns. Overall, since many profiles are presented at once, the conjoint task can be accomplished with less overall stress on the respondent and can quickly help judge preferences across many respondents.

| CBC Strengths | CBC Weaknesses |
|---|---|
| • Simulates how consumers buy in the marketplace.<br>• More enjoyable experience for respondents | • Must have a large sample size to account for the missing information from selecting only one concept per group<br>• Can not readily predict individuals preferences at the attribute level without advanced estimation techniques<br>• Limited number of attributes (5-6) can be modeled. |

**Table 4**: CBC-Strengths and Weaknesses

**Adaptive Choice-Base Conjoint Analysis**

In an effort to bring the benefits of CBC and adapt to more complex conjoint studies, Adaptive Choice-Based Conjoint ("ACBC") was introduced by Sawtooth Software in 2007 with software to implement the technique delivered in 2009 (Johnson and Orme 2007) (Hill and Orme 2011). The solution combines previous notions from ACA and recent research that suggests buyers make complex choices by simplifying the task upfront. Then they choose a product based on the simplification strategy (Johnson and Orme 2007). To mimic this behavior, an ACBC interview begins with an exercise to allow the respondent to Build Your Own product (BYO). Unlike other models, ACBC incorporates non-compensatory behavior directly into the model by allowing the respondent to constrain the preference space upfront. Once the respondent specifies his/her ideal product, then only product concepts that are close to this ideal product are explored as part of the study. An example of the BYO task is depicted below in Illustration 3 for a software product feature selection task with a minimum number of attributes.

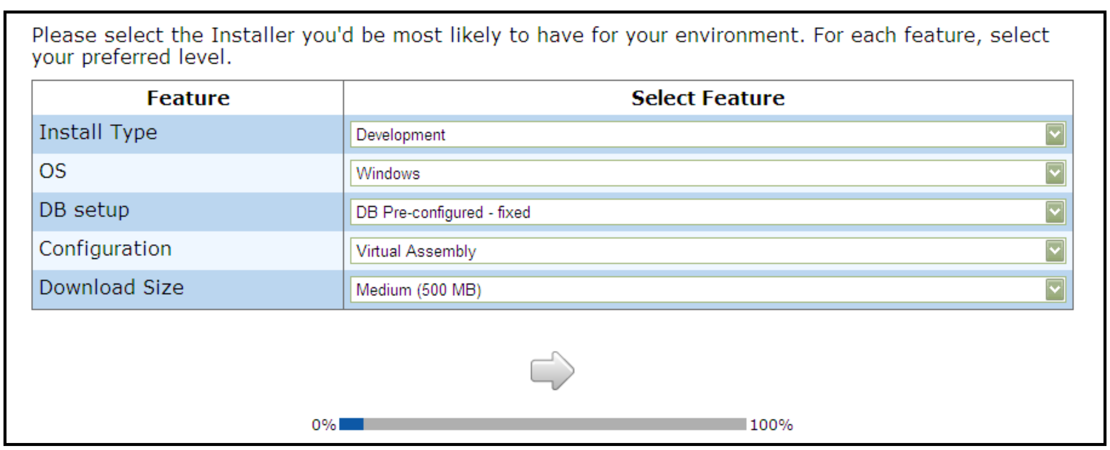
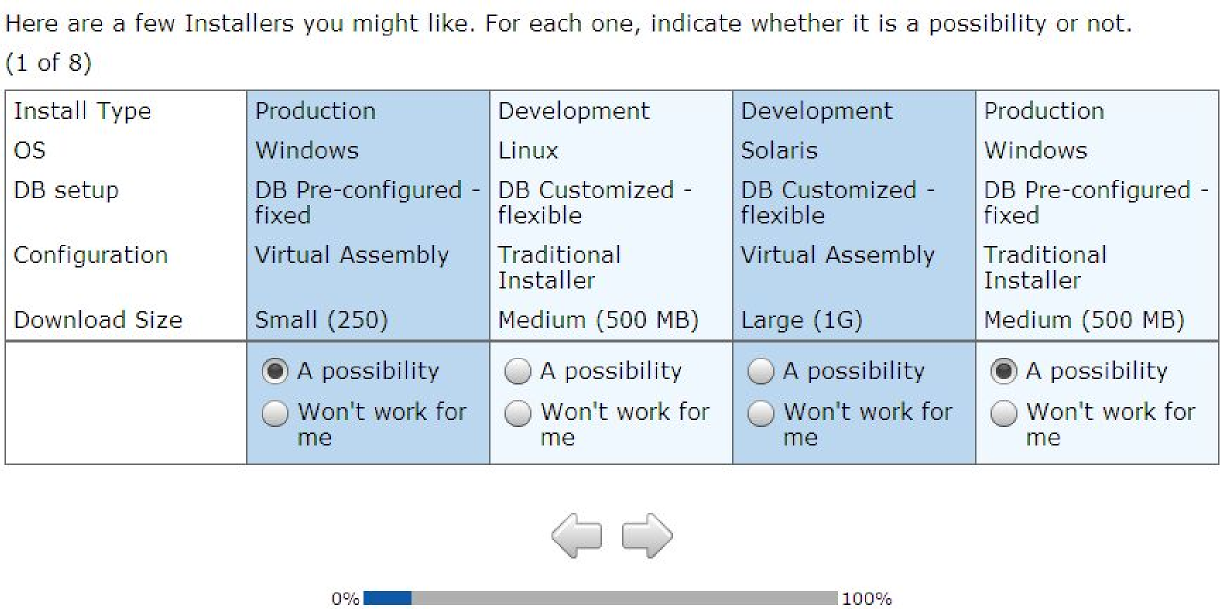



**Illustration 3**: Step 1- ACBC – Build Your Own Product

The next step of the interview process tests the respondent's willingness to trade-off some attributes levels over others, called *must-have* and *must-avoid* in the literature. This step is essentially testing the understanding of the cut-off rules the respondent is applying to evaluate the concepts by allowing him/her to specify whether the concept is a possibility or not. The interview process will ask the respondent for confirmation once it detects a must-have or must-avoid before advancing the final step of the interview process.

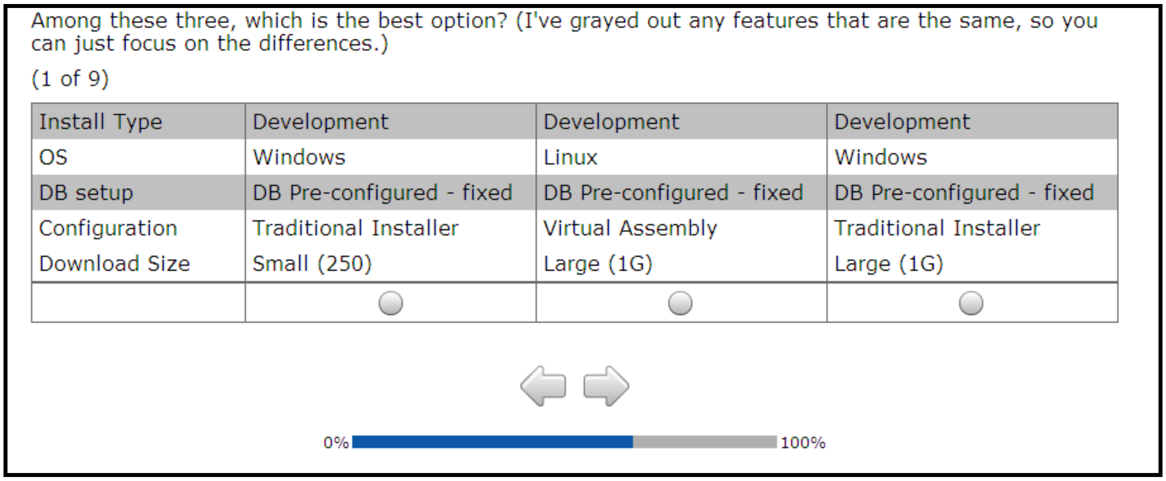

Here are a few Installers you might like. For each one, indicate whether it is a possibility or not.
(1 of 8)

| | | | | |
|---|---|---|---|---|
| Install Type | Production | Development | Development | Production |
| OS | Windows | Linux | Solaris | Windows |
| DB setup | DB Pre-configured - fixed | DB Customized - flexible | DB Customized - flexible | DB Pre-configured - fixed |
| Configuration | Virtual Assembly | Traditional Installer | Virtual Assembly | Traditional Installer |
| Download Size | Small (250) | Medium (500 MB) | Large (1G) | Medium (500 MB) |
| | A possibility / Won't work for me | A possibility / Won't work for me | A possibility / Won't work for me | A possibility / Won't work for me |



**Illustration 4**: Step 2- ACBC – Choose Possibilities

Finally, the respondent is taken through to a normal CBC like exercise; however, the only concepts shown are ones the respondent has already indicated are in an acceptable range. The attribute levels that have been deemed *must-have* are grayed out so that the respondent only needs to evaluate and trade-off on the other attribute levels. Showing the partial profile in the context of the full profile overcomes criticism cited with other partial profile approaches like ACA in the early 2000s (Alba 2003).


Among these three, which is the best option? (I've grayed out any features that are the same, so you can just focus on the differences.)

(1 of 9)

| | | | |
|---|---|---|---|
| Install Type | Development | Development | Development |
| OS | Windows | Linux | Windows |
| DB setup | DB Pre-configured - fixed | DB Pre-configured - fixed | DB Pre-configured - fixed |
| Configuration | Traditional Installer | Virtual Assembly | Traditional Installer |
| Download Size | Small (250) | Large (1G) | Large (1G) |



**Illustration 5**: Step 3- ACBC Steps – Select Product Profiles

While ACBC is relatively new, preliminary trials and research indicate quite positive results with the technique. Even though the interview process itself can be up to three times longer than normal CBC, respondents report the interview as more engaging and realistic (Orme 2010). One major benefit is that more information is gathered than in CBC exercises. Therefore, smaller sample sizes are required to stabilize the results. ACBC can model 5 and more attributes quite easily due to its ability to show partial profiles, allowing more complex products to be modeled. ACBC is already becoming quite popular with over 7 percent of conjoint studies now using this technique while having only been on the market for one year commercially (Hill and Orme 2011).

| ACBC Strengths | ACBC Weaknesses |
|---|---|
| • Many of the benefits of CBC with a smaller sample size required<br>• Handles 5+ attributes<br>• Pricing easy to measure<br>• More interaction so better respondent reaction | • Survey 2-3 times longer than comparable CBC<br>• More complex to design<br>• Too complex for small-attribute studies (4 or fewer) |

**Table 5**: Adaptive CBC - Strengths and Weaknesses

**Maximum-Difference Scaling**

Maximum difference scaling, also known as MaxDiff, is *technically* not a CA method; however, it shares the common notion of forcing consumer trade-off decisions to measure preference. The technique was introduced in the early 1990s by Jordan Louviere, its inventor, under the name of *best-worst* scaling and has experienced a surge in popularity in the conjoint measurement field (Sawtooth 2007). The percentage of researchers employing MaxDiff went from 8 to 31 percent in only a few years (Orme 2010). One of the reasons for its rapid adoption by researchers is its simplicity to measure importance from a list of multiple items. Studies ranging from 8-50 items are common in the literature (Sawtooth 2007).

Respondents seem to like MaxDiff questions as well because they are simple to understand and, like CBC, involve making choices rather than expressing the strength of preference through numerical rating and ranking. Appendix A, Figures 5 and 6 provide examples of typical survey questions for software products. Since MaxDiff is a choice technique, the same preference estimation models can be applied that are used with CBC and Adaptive CBC studies. In summary, researchers with minimal experience in statistics can conduct sophisticated studies without the design complexities or attribute limitations found in the other CA methods.

How does MaxDiff work? Respondents are shown sets of items usually 4-5 at a time. A set of items organized into a panel is typically referred to as a *task*, like in CBC. From each task, the best and worst item is chosen from the list. Each task varies the items presented and repeats items so they are shown alongside different items. An example is depicted in Illustration 6 below.

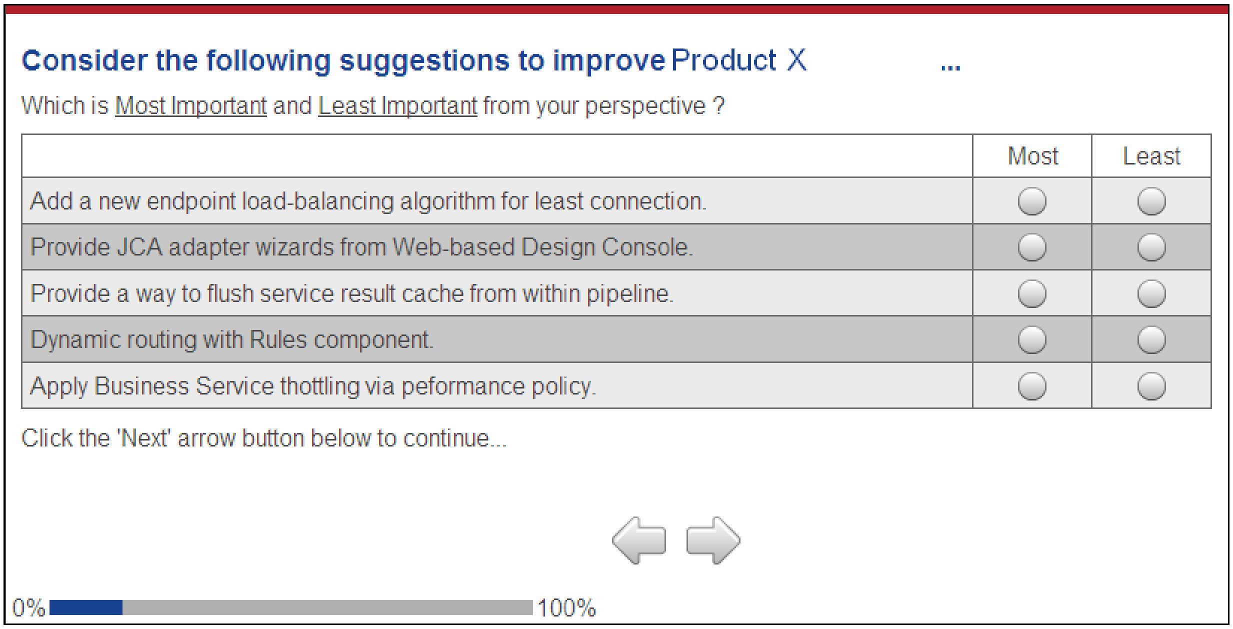



**Illustration 6**: MaxDiff Design for Software Feature Selection

To be statistically accurate, items need to be shown an equal number of times during the interview. Also, items much be shown in sufficient quantity so the item appears an equal number of times with every other item. What occurs through the interview process is an implicit ordering of items through indirect ordering e.g. **A>B** and **B>C** therefore **A>C**. Software tools that implement MaxDiff will report on design efficiency to show one-way and two-way frequencies of items with other items to guide how many tasks should be presented given the number of items in the list.

Of all the conjoint measurement techniques studied, MaxDiff holds the most promise for immediate implementation to solve common software product management tasks. Product managers routinely gather suggestions for new product features from customers and, to plan each new release, they must prioritize the list of items for engineering to implement. Minor software releases typically include about 20-40 new features or enhancements on average, depending on the complexity for engineering to implement. MaxDiff can be easily employed to measure importance of those suggestions to select new software features for product line extension and allow customers to give direct preference feedback on the most critical areas needed improvement in the software product.

| MaxDiff Strengths | MaxDiff Weaknesses |
|---|---|
| • Easy to design survey<br>• Easy to respond to survey<br>• Better than rating or ranking exercises<br>• Can be used for lists of items 50+ | • Large sample sizes are ideal<br>• Resulting model is not additive (cannot aggregate part-worths to find overall utility of different items) |

**Table 6**: MaxDiff - Strengths and Weaknesses

### PREFERENCE ESTIMATION

The field of conjoint study itself has been both thwarted and accelerated by the limitations and advancements of the available algorithms to estimate preferences. Recall that the slow adoption of CBC was not due to CBC itself. Researchers actually preferred the interview technique. Rather, the estimation algorithm available, aggregate logit, was not robust enough to estimate individual utilities from the sparse choice data.

Estimation models were invented or borrowed from other fields to overcome limitations and then only adopted by researchers when computer software and hardware advanced to make their use practical (Johnson 2005). Through the 1970's and 1980's, estimation was dominated by linear regression analysis. In the late 1990s, the introduction of HB and LCA estimation allowed a new level of sophistication to be achieved with sparse data sets of CBC, ACBC and MaxDiff. Table 7 below summarizes the most common techniques leveraged to estimate preference data.

| Estimation | Purpose | Software Tools |
|---|---|---|
| Multiple Linear Regression<br>Logit (All) | Aggregate Utilities/Partworths<br>(Interval Data) | • Excel<br>• Sawtooth Software<br>• Survey Analytics<br>• SPSS<br>• SAS |
| Counting Analysis<br>(CBC, MaxDiff) | Aggregate Utilities<br>(Ratio Data) | • Sawtooth Software |
| Importance Analysis<br>(CA, ACA) | Attribute Importance | • Excel<br>• Sawtooth Software |
| Latent  Class | Segmentation<br>Subgroup Utilities | • Sawtooth Software<br>• Latent Class GOLD |
| Hierarchical<br>Bayes | Individual and<br>Aggregate<br>Utilities/Partworths | • Sawtooth Software |

**Table 7**: Preference Estimation/ Analysis Methods[8]

Once preference data are collected, estimation algorithms are applied and the results analyzed with different software tools to help with interpretation.  Depending on the questions the preference study seeks to answer, different estimation techniques are employed.  The estimation algorithms should be selected in the concept phase of the preference measurement study as there are benefits and potential drawbacks to each outlined in next sections.  For example, if you seek to understand preferences among different subgroups in your respondent base, you cannot discover this using aggregate estimation.  You will need to employ Latent Class estimation for this task.

[8] The list presented here is not exhaustive, rather the most readily available tools discovered while researching this paper.

**Aggregate Analysis – Multinomial Logit**

The simplest preference measurement technique is multiple linear regressions. Traditional CA fits this model well because, by definition, it's essentially a multiple linear regression problem with the dependent variable being respondent's ratings of product profiles (Orme 2010). The calculated coefficients of the independent variables are the partworth utilities of the product. Typically least squares and logit functions are used for the regression function.

Aggregate analysis is highly effective when combined with traditional CA and ACA data collection techniques because these methods estimate rich information at the individual level that can be averaged to find group preferences and other trends of the data. Aggregate logit is not as valuable for choice-based studies such as CBC, ACBC and MaxDiff since it will find average utilities across the entire respondent pool. Not enough information is captured by choice-based methods to reveal individual level partworth utilities. Aggregate is fine if one assumes a homogenous respondent population; however, it is quite problematic if measuring preferences for a heterogeneous respondent pool. For example, if half the respondents favor one level of attribute and the other half another, the average utilities would cancel each other out entirely, showing the utility of that attribute to be zero.

Aggregate analysis was the primary way to estimate conjoint measurement studies through the 1970's, 80's and 90's. It is easy to implement with Excel and readily available in most software packages that provide statistical algorithms. It has limited usefulness for choice-based data unless under pristine data collection conditions.

**Latent Class Analysis**

Latent Class Analysis ("LCA") is an estimation method used to find ad-hoc or *latent* segments that may exist in a data set. LCA is a statistical method, calculating partworth utilities for potential subgroups. Rather than a pure mathematical model, LCA assigns a probability that a given respondent is a member of a subgroup or class. Once subgroups are identified, one typically wants to understand what preferences define this subgroup and perhaps label it for further investigation with market simulators or make product decisions to appeal to that class profile of respondent.

The number of subgroups can be fine-grained or course-grained depending on the goals of the study, and the rules for determining the best grouping relies on statistical rules of thumb such as looking at the AIC – Akaike information criterion or other statistical indicators of algorithm stabilization (Cohen 2003).

LCA can be used with a variety of data. It became popular in the mid-1990s as a tool for analyzing CBC data sets because the model typically provided more insight about the structure of respondent preferences than aggregate logit (Cohen 2003). During the late 1990s and through today, the use of HB for modeling CBC data has eclipsed that of LCA in terms of popularity. However, latent class remains a valuable tool as it provides the benefits of aggregate estimation while performing segmentation for choice data, recognizing the heterogeneity of the respondent pool.

**Hierarchical Bayes**

HB estimation is a statistical analysis method that has become prominent as it allows for study of high-dimensional data and complex relationships that are common in marketing (Rossi an Allenby 2003). While the details of HB are beyond the scope of this paper, these methods provide better ways to estimate aggregate and individual level part-worth utilities while revealing heterogeneity of the respondent pool with sparse sets of data like those provided with CBC, ACBC and MaxDiff data collection techniques. HB offers a way to borrow information from each respondent to stabilize the distribution curve of individual's part-worth estimates. HB has become the de facto method for analyzing choice-based data sets.

### EMERGING

There are new methods on the horizon for preference measurement, e.g., Analytical Hierarchy Process ("AHP") and artificial neural networks ("ANN"). AHP has been leveraged in several recent empirical studies and proven to be effective in complex product evaluation tasks. This ability to deal with complexity may help its application to complex software product tasks. AHP has shown results commensurate with those performed with CBC and HB estimation (Meibner and Decker 2009). . However, this research is still very new and no commercial software yet exists to apply AHP outside research environments. One promising claim is that it appears to improve respondent fatigue by minimizing attribute level combinations.

Research is also being conducted to apply non-linear estimation for predicting choice decisions with artificial neural networks ("ANN"). The hope for ANN is that it may be able to better account for attribute interactions and detect hidden co-dependencies amongst attributes without the rigid rules regarding orthogonality now required in linear CA methods (West et. al 1999). Studies have shown performance to be at least as good for predicting choice as CBC and HB estimation; however, like AHP, no commercial software is yet available to implement ANN for preference measurement tasks.

### SOFTWARE TOOLS

An example estimation leveraging Excel is provided in Appendix A, Figures 1-4. These figures capture the process for performing multilinear regression for a study with 3 attributes to model preference for software feature using traditional CA. In Excel, attributes are encoded as variables into matrices and then dummy coded to perform multiple linear regressions to estimate coefficients to reveal partworths of the attribute levels and overall utility of the attributes. Example output of the analysis is captured in Appendix A, Figure 1 along with calculated attribute importance.

While simple preference estimation and analysis can be done in Excel, most studies will need to employ commercial software for sophisticated analysis. Sawtooth Software is recognized as the leader in providing CA software for preference study design, data collection and estimation. Sawtooth Software was founded by one of the pioneer practitioners, Ralph Johnson, and is widely leveraged by market researchers (Johnson 2005). Sawtooth offers analysis modules for all major estimation techniques as well as tools to design surveys for data collection for all major conjoint methods, including MaxDiff. Sawtooth provides market simulators to conduct what-if analysis once utilities are derived from the preference data. Market simulators can help with what-if analysis for new concept testing or pricing/brand research. Another notable package is Survey Analytics. This software is Web-based and allows design tools for CBC and MaxDiff studies and data collection; however, the only estimation module available as of this writing is aggregate analysis. If the preference study is only trying to find basic trends, and one can assume that the respondent base is fairly homogenous, Survey Analytics offers a nice Web-based alternative to more sophisticated software provided by Sawtooth Software.

## Chapter 3: Empirical Study

There are three key areas of software product management that could potentially benefit from consumer preference measurement studies– software product improvement, line extensions and product concept testing. For mature software products, new concept testing is usually done to improve a product vs. building a new product from scratch. An example might be providing an easier way to configure or install the software product or adding a new user interface for the software. Product improvement and line extension tasks typically involve selecting and prioritizing a set of suggestions from customers for enhancing existing features or adding new ones for the next release of the product. Today, rating style and chip allotment surveys are sometimes engaged to help with these areas with limited quantitative value. Refer to Appendix A, Figures 5 and 6 for examples of rating and chip allotment survey questions, respectively.

The goal of the empirical study was to select and apply a preference measurement technique and derive actionable results and decisions from its application. To help with the empirical study, a framework was adopted from the literature to help map software product management's tasks to preference measurement techniques as outlined below (Netzer et. al 2007).

1. Identify the problem the preference study will address.
2. Establish the questions the study will answer.
3. Design the preference measurement task and data collection approach.
4. Estimate preference data.
5. Convert the results into actionable measures.

## IDENTIFY THE PROBLEM

. A common task for a software product manager is to prioritize suggestions for product improvements for each release of the product. Suggestions for improvement come from various sources and it can be a difficult task to prioritize across the entire customer base. The problem the preference measurement study should solve is injecting direct customer feedback across the suggestions to help prioritize. This information can guide product management decisions and ensure investments are being made where they matter most to customers.

### Background

Software product managers typically engage with customers to elicit feedback and bring the customer closer to the product design and development process. Customer visits and focus groups are organized to solicit feedback on how to improve the software product. These interactions tend to be qualitative in nature, although rating and chip allotment surveys are sometimes engaged to organize the feedback. Product roadmaps are routinely shared with discussion sessions where customers give feedback verbally or may provide wish lists of new features to solve certain problems they face. Customers may also engage with the customer support organization and communicate enhancements through this channel. These suggestions are sent along to product management for consideration.

Suggestions are typically logged into software feature tracking software and evaluated based on product management perspective of what problem the suggestion will solve. Over the course of many customer visits and focus groups, product enhancement suggestions can become quite difficult to prioritize for the next version of product. Many times, all improvements suggested may be important from a particular customer's perspective while others may only be important to a select group and conflict with the interests of others.

The type of software product chosen for the empirical study is an *integration* software product. Integration software is defined as software that allows one to combine or *integrate* other software together to form new applications for the customer's business. As such, the software tends to be quite technical. The consumer of the software product is typically a software programmer with very detailed wants and needs. Thus, the features discussed with these types of customers tend to be quite specialized and not very user-friendly unless one is an expert at integration software. Modeling a survey on such complex concepts can be quite difficult and complicates the mapping of consumer preference techniques since the respondent may already be overloaded by many attributes, product concepts and profiles.

Software product managers today routinely employ rating style or chip allocation surveys to help rate and rank new feature requests and suggestions (Appendix A, Figures 5 and 6 are examples of this style of survey). While on the surface, these types of surveys seem to provide quantitative data regarding preferences, ratings style surveys have been proven in research to suffer from both scale bias and cultural bias (Cohen 2003). Scale bias is the tendency to use the rating scale in different ways i.e. mark everything important or not unimportant. More recently, evidence has surfaced to show that scales do not translate well across cultures, one culture using a scale of numbers differently than another culture.

Chip allocation surveys seem more related to preference measurement techniques than straight ratings, as at least the respondent is being asked to choose to spend a limited resource (chips) in an effort to force a trade-off decision amongst competing features. In practice, this often leads to inconclusive results. For example, in research for this paper, a typical chip allotment task was analyzed. When the respondents were asked to prioritize two important suggestions for providing new software installers, the respondents often allocated 50/50 or 60/40 to each suggestion. The results of the survey were basically a tie, giving no information to help guide product management which item was a higher priority overall. See Appendix A Figure 6 for the split results of the chip allotment task..

### ESTABLISH THE QUESTIONS

For this study, the important questions to be answered in the study were

- What are the most important features/suggestions that we should include in the next release of our product? We would like to select those features with the biggest impact across a wider customer base.
- What features/suggestions should be dropped entirely from consideration since so few customers are interested in them? These would not be a good investment of engineering resources since they are of so little value to the broader customer base.
- Are there ad-hoc segments in the customer base that care about certain features/suggestions more than others? Can we capitalize on these segments if we bundle certain high priority features together to appeal to these ad-hoc segments?

### DESIGN PREFERENCE MEASUREMENT TASK

MaxDiff was selected for this study because it excels in allowing one to rate the importance of a list of unrelated items to determine priority. This almost exactly matches the task most software product managers face when reviewing suggestion for new product releases. Typically, new releases of software products are incremental changes. The new features are usually minor in scope; however, there are always too many suggestions than can possibly be included in a minor release hence the need to select those that will benefit the majority of the customer base and not just a select few. There are always exceptions to the rules, of course, for long-standing, major customers.

To implement the study, a list of over 30 product enhancement suggestions were extracted from the software requirements software that were currently being tracked for the next minor release of the product. Descriptions of features ranged from 5-12 words. Suggestions were reviewed by product management with engineering for completeness. A few concepts were added as test concepts. These items are planted to test the validity to the MaxDiff results.

The MaxDiff survey was designed to leverage SSI Web from Sawtooth Software. A few decisions had to be made, e.g. how many product feature suggestions to show respondents per task and how many tasks overall to a statistical efficient study. Design tests were run by the software to ensure statistical efficiency. Sample surveys were delivered and measured before rolling out to a wider audience. Based on initial feedback, the MaxDiff design was modified to include a panel to encourage the respondents to keep going at the half way mark. Example panels from the MaxDiff data collection are shown in Appendix A, Figures 7-9. The actual design of the survey only took a few days. The design set up for delivery to respondents took about one week, mostly due to logistics. Statistics of the survey design are shown below.

```
Total number of tasks: 2310
Average tasks per respondent: 30
Average concepts per task: 5
Average attributes per concept: 23
Number of respondents: 77
```

Questions were included in the design to gauge the effectiveness of the survey type itself. The questions included in the MaxDiff survey are shown in Illustration 7.

The results were quite positive, with a majority of respondents finding the survey simple, realistic and with reasonable amount of time spent on the survey. Average time to complete the survey was 9 minutes.

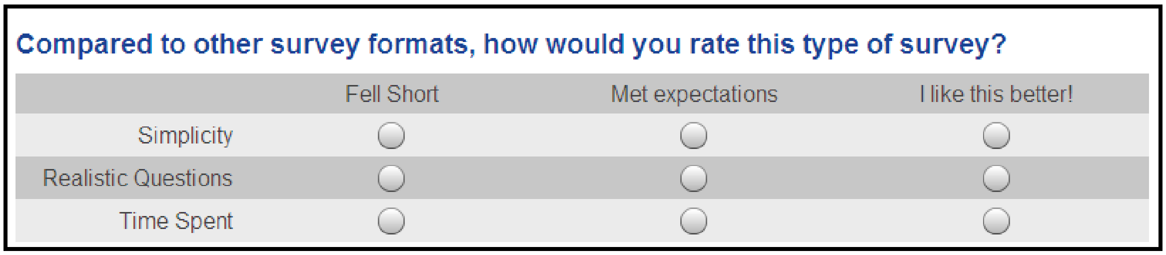



**Illustration 7**: Questions for Respondents on MaxDiff Survey Style.

### Estimate The Preference Data

MaxDiff is a choice measurement so the same preference estimation techniques employed for CBC and ACBC are available for MaxDiff data. The fastest and easiest way to get feedback from MaxDiff is to analyze the counts. Counts literally specify directly how many times a concept was selected *best* and *worst* when shown in a task panel. It gives a quite reliable indication of priority of the items or concepts at quick glance. For software prioritization, counts can quickly give a top 5 or bottom 5 list of the software features most or least wanted across a respondent pool. It gives some indication of conflicts in the respondent base but cannot help one understand them further. For this deeper analysis, one needs to employ LCA for segmentation or HB to look at individual preferences.

Illustration 8 shows the counts for features selected best, plotted alongside how many times the feature was selected worst from a task, ordered by best. Notice the strong reaction to feature 21 below. This item was selected worst almost double the amount of times of any other feature; however, in some contexts it warranted as the best selection. This could be an indicator of a latent class that desires this feature; however, others in another group (larger one) do not desire this feature.

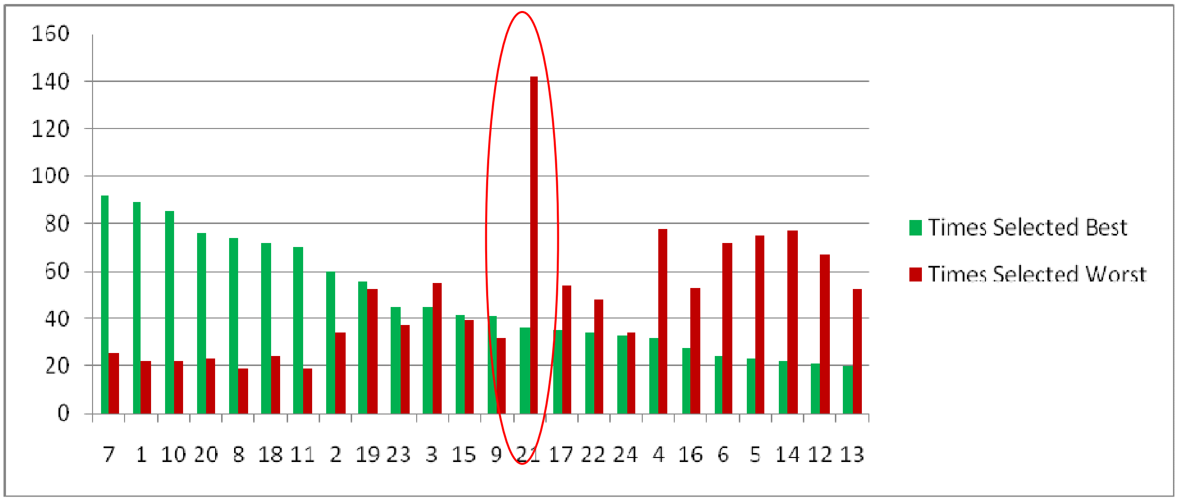



**Illustration 8:** MaxDiff Counts for Best and Worst

Choice preference measurements, like MaxDiff tend to require larger sample sizes of respondents since the data is sparser than that collected from individual rating style data as in traditional CA or ACA. The estimation techniques, however, have proven to provide excellent ability to estimate individual scores from the sparse data. For analyzing trends, a respondent pool of 77 from 200 was considered sound. If the study was going to employ market simulation to analyze utilities for maximizing profit or competitive products, a slightly larger sample size would be recommended.

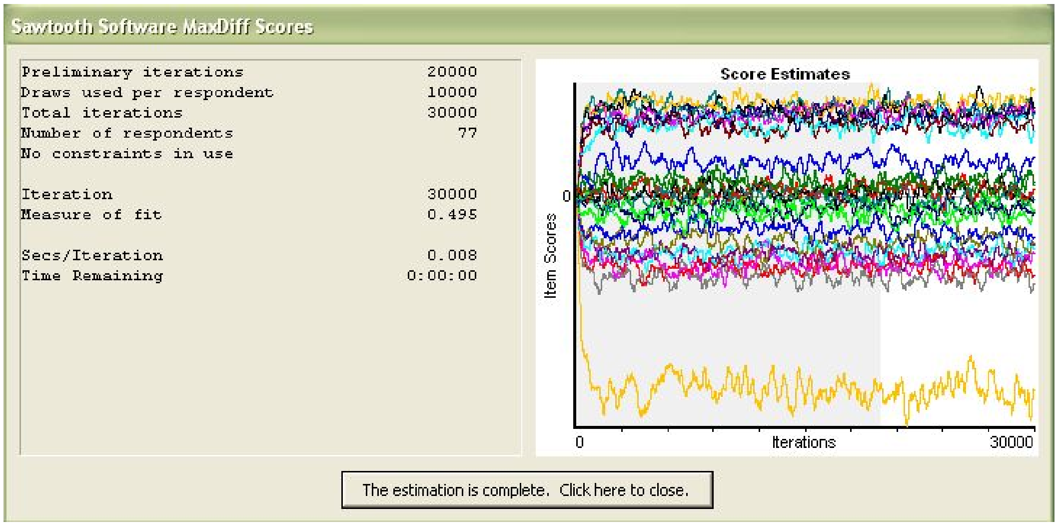



**Illustration 9**: Hierarchical Bayes Estimation

To estimate the attribute importance and utilities, HB estimation is run on the data. While HB runs, a graphic is shown that characterizes how well the estimated scores are stabilizing over the planned thousands of iterations. The graphic plots the estimated scores for each iteration of the process. The estimates all start at 0, and then trend toward their final values. Once the process has "converged" the estimates will tend to randomly "wobble" up and down, but there should be no noticeable trend remaining.

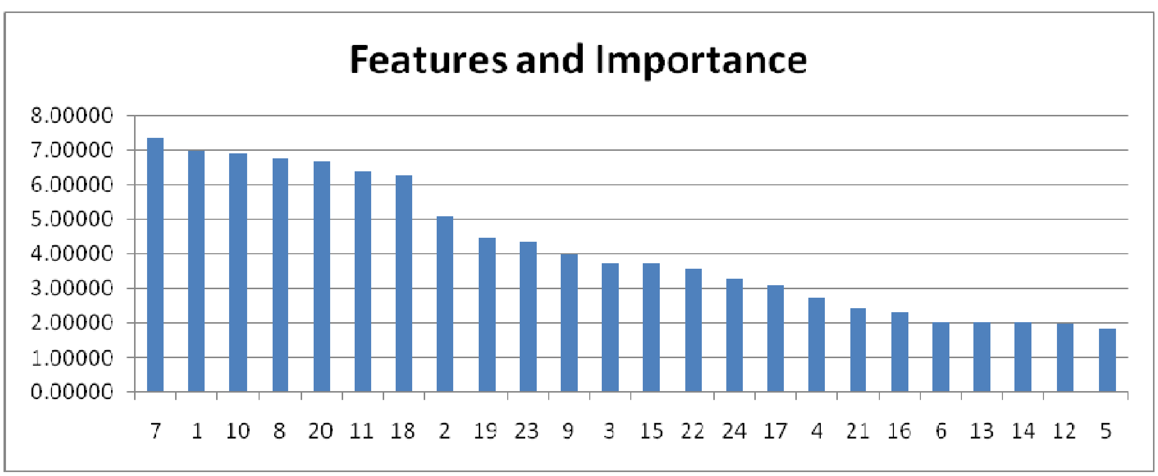



**Illustration 10:** Feature Importance from HB Estimation

After reviewing the overall importance of the suggestions, Latent Class analysis was run to uncover potential subgroups with difference preferences stated within the group. The results are quite interesting and are shown in Illustration 10. Two subgroups clearly emerged with different preferences on several key feature enhancements. Further investigation will be done to label and profile these subgroups to allow the product to align along these profiles. One profile seems to align with pure developer interests while another aligns with software production interests.

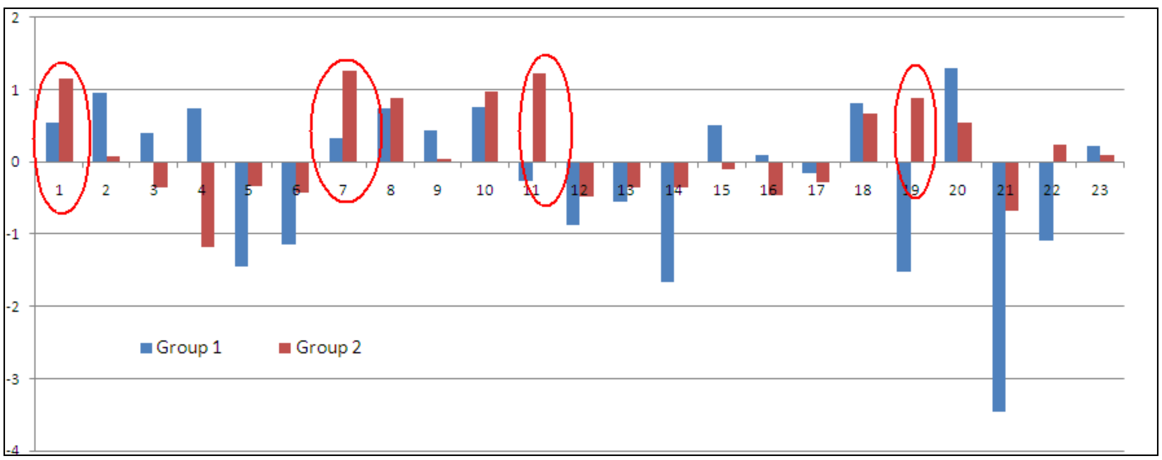



**Illustration 11:** Latent Segments and Preferences

### Conversion Into Action

The results of the empirical study were quite satisfying. With relative ease, all the suggestions for the next minor release of product were prioritized over a heterogeneous respondent pool. While theories existed that there was a divisive set of interests generating these suggestions, the collected preference data with Latent Class analysis allowed us to examine this further and define ad hoc segments for further investigation. The data suggests up to five subgroups may exist, differentiated by their preferences. However, the next product release will try to capture the most predominant preferences identified in the 2 group segmentation. The top features are now committed, and evaluations of the features designated worst will be further evaluated from the study.

MaxDiff worked extremely well for this empirical study and met expectations of what information might be gleaned from the study. The respondents seemed to enjoy the survey overall and appreciated the more direct approach to gather feedback for product improvement. An implicit assumption was made during the survey design that the respondents would be seasoned product expert. If less seasoned respondents would have been included in the study, explanations of all the suggestions would have been required. Even then, one would worry about fatiguing this type of respondent with too many technical concepts. In summary, if you must solicit preferences from different skill levels as part of the study, you may need to consider alternate designs to appeal to different respondent groups to achieve the goals of the study.

# Chapter 4: Conclusions

This paper recommends using preference measurement techniques for guiding software product management decisions. The methods for preference measurement have consistently improved over time and now there are many choices of studies and estimation techniques. Given the variety, it can be difficult for practitioners and researchers to decide which method is appropriate or best for their situation. The good news is that most of the preference measurement methods, whether they be traditional CA, CBC, ACBC or MaxDiff, deliver good results within their prescribed limitations (Orme 2010). That said, choosing which method is best for a particular application can be a daunting task if one does not use the methods on a daily or weekly basis. Table 8 below quickly summarizes the most relevant methods for application for software product management tasks.

◉ Recommended for Software Product Managers

| Preference Design & Data Collection | Preference Estimation | | |
|---|---|---|---|
| | Aggregate / Multilinear Models | Hierarchical Bayes | Latent Class Analysis |
| Traditional Conjoint Analysis | ✓Individual Importance<br>✓Aggregate Importance<br>✓Predictive Market Choice<br>✓Brand/Pricing | N/A | N/A |
| Adaptive Conjoint Analysis | ✓Individual Importance<br>✓Aggregate Importance<br>✓Predictive Market Choice | ✓Individual Importance<br>✓Aggregate Importance<br>✓Predictive Market Choice | N/A |
| Choice-Based Conjoint Analysis | ✓Aggregate Importance | ✓Individual Importance<br>✓Aggregate Importance<br>✓Brand/Pricing<br>✓Predictive Market Choice ◉ | ✓Ad-hoc Segmentation<br>✓Predictive Market Choice – Subgroup ◉ |
| Adaptive Choice-Based Conjoint Analysis | ✓Aggregate Importance | ✓Individual Importance<br>✓Aggregate Importance<br>✓Brand/Pricing<br>✓Predictive Market Choice ◉ | ✓Ad-hoc Segmentation<br>✓Predictive Market Choice – Subgroup ◉ |
| Maximum Difference Scaling | ✓Aggregate Importance | ✓Individual Importance<br>✓Aggregate Importance ◉ | ✓Ad-hoc Segmentation<br>✓Predictive Market Choice – Subgroup ◉ |

**Table 8:** Summary of Preference Design, Data Collection and Estimation.

To conduct a successful preference management study, a product manager needs to have an understanding of the estimation techniques and the questions they can be employed to answer. Based on the empirical studies detailed in this report, the techniques in the Table 8 above appear best suited for solving the most common problems the software product manager faces on a routine basis – software feature prioritization for product improvement, product line extension and new concept testing for new capabilities of the software.

As demonstrated in Chapter 3, MaxDiff is an effective tool for software product improvement tasks. The survey style was liked and well-received by respondents and the data gleaned was quite valuable for making product management decisions. MaxDiff is a simple-to-design conjoint measurement technique, and this is reflected in the software tools that implement it. Adaptive CBC is highly effective for software product configuration or concept testing, provided a reasonable and valid model can be derived from 7-9 attributes with levels approaching the same scale.

To apply preference measurement to software product management, a deep understanding of the product is required to ensure sound data collection designs, principles of orthogonality and main effects. This implies that the design of the study itself cannot easily be out-sourced to market researchers as recommended by much of the CA literature. One can imagine a process where the CA software captures more expertise to guide the product manager for preference design and data collection. Then a partnership with market researcher for the estimation and interpretation is still valid.

More work is needed in general to simplify the preference measurement literature and software targeted at product management professionals vs. pure market researcher and practitioners. The software for CA measurement, outside of MaxDiff, needs to appeal to non-statisticians to be widely adopted. Sawtooth Software is approaching this level of simplicity with its MaxDiff software module in SSI Web. Survey Analytics, while extremely simply to use, does not provide the robust data analysis that is likely required to glean the most from preference measurement study.

In closing, this paper recommends the use of consumer preference measurement for software product management decisions as outlined in the Table 8 above. As demonstrated in Chapter 3, these advanced marketing tools can be easily incorporated into software management practices and ensure investments are made where they matter most to customers for product improvement. Injecting customer feedback directly into the product concept and development phases to guide decisions can help build brand loyalty and differentiate in a highly competitive marketplace.

# Appendix A

**Summary**

| Configuration and Deployment | Average | Range |
|---|---|---|
| Eclipse-based Tooling | 0.00 | |
| Parameterized Proxy | 0.54 | 0.54 |
| **Monitoring and Reporting** | | |
| Adaptive Routing | 0.00 | 1.72 |
| Identity-based metrics | -1.72 | |
| Amberpoint - Partner Certification | -0.44 | |
| **Connectivity & Standards** | | |
| Outbound RDBMS | 0.00 | 1.69 |
| Custom Security Credentials | -0.39 | |
| WS-Reliable Messaging- OneWay | -1.69 | |

**Aggregate Analysis - Attributes Overall Importance Relative to Other Attributes**

| Configuaration and Deployment |
|---|
| 13.60% |

| Monitoring & Reporting |
|---|
| 43.61% |

| Connectivity Standards |
|---|
| 42.79% |

**Figure 1**: Attribute Importance with Aggregate Analysis

| | | | | Reference | | | |
|---|---|---|---|---|---|---|---|
| Card | Configuration &Deployment | Monitoring and Reporting | Connectivity & Standards | Card | Configuration&Deployment | Monitoring and Reporting | Connectivity & Standards |
| 1 | 1 | 1 | 1 | 1 | Eclipse-based Tooling | Adaptive routing | Outbound RDBMS |
| 2 | 1 | 1 | 2 | 2 | Eclipse-based Tooling | Adaptive routing | Custom Security Credentials |
| 3 | 1 | 1 | 3 | 3 | Eclipse-based Tooling | Adaptive routing | WS-Reliable Messaging- OneWay |
| 4 | 1 | 2 | 1 | 4 | Eclipse-based Tooling | Identity-based metrics | Outbound RDBMS |
| 5 | 1 | 2 | 2 | 5 | Eclipse-based Tooling | Identity-based metrics | Custom Security Credentials |
| 6 | 1 | 2 | 3 | 6 | Eclipse-based Tooling | Identity-based metrics | WS-Reliable Messaging- OneWay |
| 7 | 1 | 3 | 1 | 7 | Eclipse-based Tooling | Partner Certification | Outbound RDBMS |
| 8 | 1 | 3 | 2 | 8 | Eclipse-based Tooling | Partner Certification | Custom Security Credentials |
| 9 | 1 | 3 | 3 | 9 | Eclipse-based Tooling | Partner Certification | WS-Reliable Messaging- OneWay |
| 10 | 2 | 1 | 1 | 10 | Parameterized Proxy | Adaptive routing | Outbound RDBMS |
| 11 | 2 | 1 | 2 | 11 | Parameterized Proxy | Adaptive routing | Custom Security Credentials |
| 12 | 2 | 1 | 3 | 12 | Parameterized Proxy | Adaptive routing | WS-Reliable Messaging- OneWay |
| 13 | 2 | 2 | 1 | 13 | Parameterized Proxy | Identity-based metrics | Outbound RDBMS |
| 14 | 2 | 2 | 2 | 14 | Parameterized Proxy | Identity-based metrics | Custom Security Credentials |
| 15 | 2 | 2 | 3 | 15 | Parameterized Proxy | Identity-based metrics | WS-Reliable Messaging- OneWay |
| 16 | 2 | 3 | 1 | 16 | Parameterized Proxy | Partner Certification | Outbound RDBMS |
| 17 | 2 | 3 | 2 | 17 | Parameterized Proxy | Partner Certification | Custom Security Credentials |
| 18 | 2 | 3 | 3 | 18 | Parameterized Proxy | Partner Certification | WS-Reliable Messaging- OneWay |

**Figure 2**: Traditional CA Attribute-Level Encoding in Excel

| Card | Eclipse-based Tooling | Parameterized Proxy | Adaptive Routing | Identity-based Metrics | Amberpoint Certification | Outbound RDBMS | Custom Security Credentials | WS-Reliable Messaging |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 2 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 3 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 4 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 5 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 6 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 7 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 8 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 9 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 10 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 11 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 12 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 13 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 14 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 15 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 16 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 17 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 18 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |

**Figure 3**: Full Dummy Coding for Traditional CA in Excel

| Card | Parameterized Proxy | Identity-based Metrics | Partner Certification | Custom Security Credentials | WS-Reliable Messaging | Preference on Scale 1-10 |
|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 9 |
| 2 | 0 | 0 | 0 | 1 | 0 | 9 |
| 3 | 0 | 0 | 0 | 0 | 1 | 6 |
| 4 | 0 | 1 | 0 | 0 | 0 | 7 |
| 5 | 0 | 1 | 0 | 1 | 0 | 6 |
| 6 | 0 | 1 | 0 | 0 | 1 | 5 |
| 7 | 0 | 0 | 1 | 0 | 0 | 9 |
| 8 | 0 | 0 | 1 | 1 | 0 | 9 |
| 9 | 0 | 0 | 1 | 0 | 1 | 7 |
| 10 | 1 | 0 | 0 | 0 | 0 | 10 |
| 11 | 1 | 0 | 0 | 1 | 0 | 10 |
| 12 | 1 | 0 | 0 | 0 | 1 | 6 |
| 13 | 1 | 1 | 0 | 0 | 0 | 7 |
| 14 | 1 | 1 | 0 | 1 | 0 | 8 |
| 15 | 1 | 1 | 0 | 0 | 1 | 6 |
| 16 | 1 | 0 | 1 | 0 | 0 | 10 |
| 17 | 1 | 0 | 1 | 1 | 0 | 10 |
| 18 | 1 | 0 | 1 | 0 | 1 | 8 |

**Figure 4**: Partial Encoding to Perform Multiple Linear Regression in Excel

*With dummy coding, one level is dropped from each attribute to hold it constant so the equations can be optimized to find coefficients of other levels using multiple linear regressions.*

.

| Routing Messaging and Transports | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| Dynamic message routing based on content from incoming message | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |
| Synchronous routing | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |
| Asynchronous routing | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |
| Bridging of synchronous and asynchronous business services | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |
| Messaging to business services on ESB through transport protocols. | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |
| JMS | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |
| HTTP(S) | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |
| FTP(File) | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |
| Drop | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |
| Email(SMTP) | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |
| JMS | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |
| MQ Series | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |
| Dynamic callout in message flow to enrich message and/or lookup message routing information | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |

**Figure 5**: Typical Rating Type Survey for Software Feature Importance

**2. You have $100 to spend on installers. How would you allocate your money?**

| | Response Average | Response Total | Response Count |
|---|---|---|---|
| All-encompassing installer: supports all topologies and configurations. Large, all possible options. | 46.29 | 1,296 | 28 |
| **"1-click" Developer install: assumes standalone server and default dev settings. Smaller, minimal options.** | **53.71** | **1,504** | 28 |
| | | **answered question** | **28** |
| | | **skipped question** | **0** |

**Figure 6**: Typical Chip Allotment Survey Results

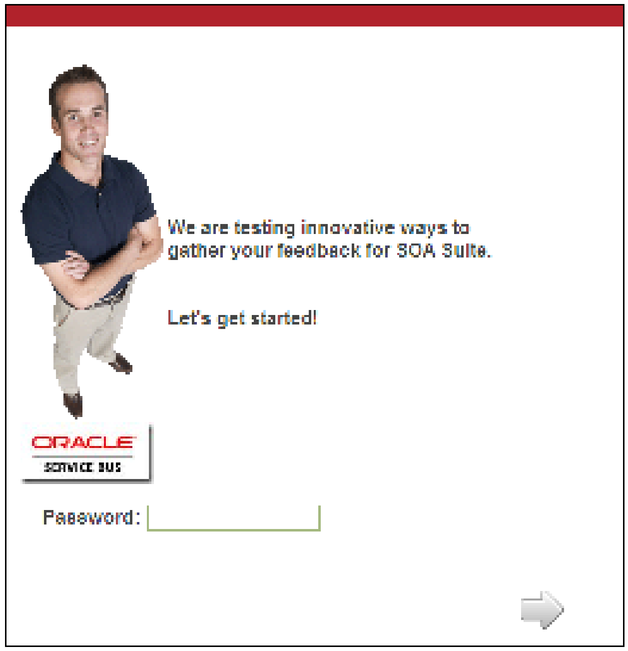



**Figure 7**: Introduction to the MaxDiff Survey

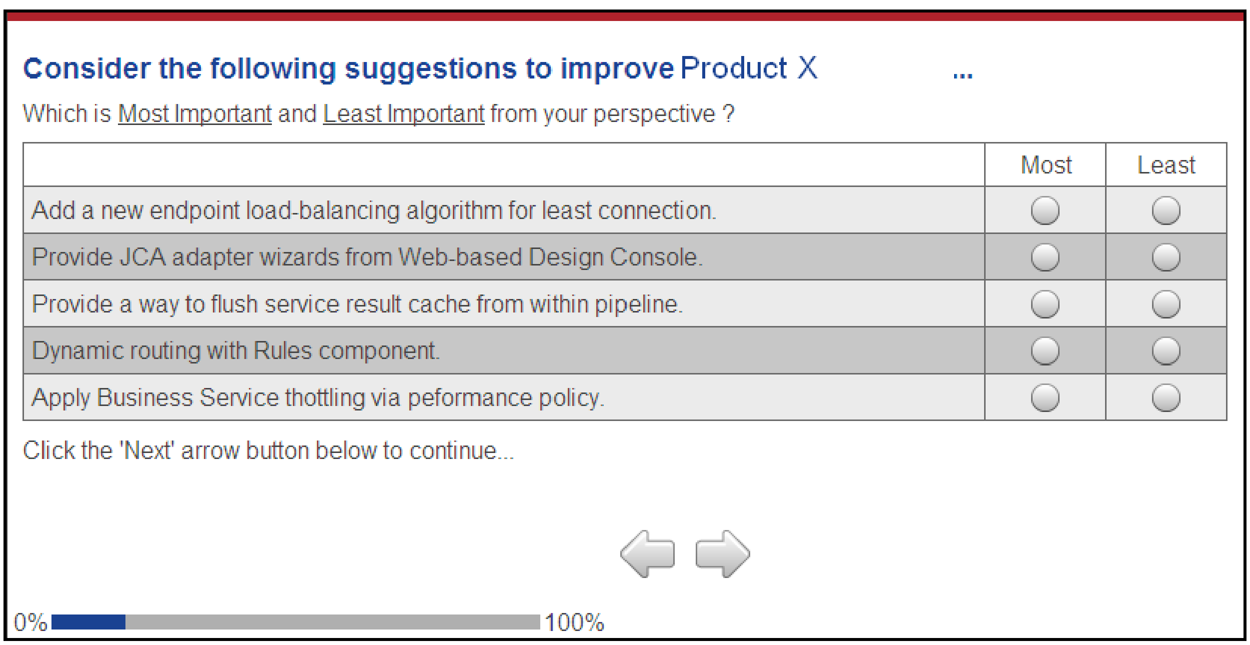



**Figure 8**: Sample Panel for Software Feature Selection


**Figure 9**: Encouragement during the MaxDiff Survey

**MaxDiff Counts Analysis**

| | |
|---|---|
| MaxDiff Exercise | requirements |
| Sets Included | All |
| Respondents Included | Completes |
| Total Number of Respondents | 77 |
| Total Number of Sets | 1155 |

| Label | Item Number | Times Shown - Best | Times Selected Best | Best Count Proportion | Times Shown - Worst | Times Selected Worst | Worst Count Proportion |
|---|---|---|---|---|---|---|---|
| Virtualize services onto OSB f... | 1 | 239 | 89 | 0.372 | 239 | 22 | 0.092 |
| Allow role-based access to pr... | 2 | 244 | 60 | 0.246 | 244 | 34 | 0.139 |
| Add support for multiple OSB ... | 3 | 243 | 45 | 0.185 | 243 | 55 | 0.226 |
| Option to "warm up" a servic... | 4 | 238 | 32 | 0.134 | 238 | 78 | 0.328 |
| Import OSB configurations fro... | 5 | 245 | 23 | 0.094 | 245 | 75 | 0.306 |
| Provide Maven plug-in for aut... | 6 | 236 | 24 | 0.102 | 236 | 72 | 0.305 |
| Provide JCA adapter wizards f... | 7 | 236 | 92 | 0.390 | 236 | 25 | 0.106 |
| Fault policy consistent with ot... | 8 | 239 | 74 | 0.310 | 239 | 19 | 0.079 |
| Apply operational settings and... | 9 | 244 | 41 | 0.168 | 244 | 32 | 0.131 |
| Apply Business Service thottli... | 10 | 243 | 85 | 0.350 | 243 | 22 | 0.091 |
| Dynamic routing with Rules c... | 11 | 244 | 70 | 0.287 | 244 | 19 | 0.078 |
| Add new generic loop constru... | 12 | 236 | 21 | 0.089 | 236 | 67 | 0.284 |
| Add a new endpoint load-bala... | 13 | 244 | 20 | 0.082 | 244 | 52 | 0.213 |
| Apply B2B transformations dir... | 14 | 240 | 22 | 0.092 | 240 | 77 | 0.321 |
| Action to easily convert from ... | 15 | 242 | 42 | 0.174 | 242 | 39 | 0.161 |
| Provide a way to flush service... | 16 | 239 | 28 | 0.117 | 239 | 53 | 0.222 |
| Allow caching at the SOAP o... | 17 | 240 | 35 | 0.146 | 240 | 54 | 0.225 |
| Provide an easy way to add n... | 18 | 241 | 72 | 0.299 | 241 | 24 | 0.100 |
| Virtual Assembly appliance for... | 19 | 244 | 56 | 0.230 | 244 | 52 | 0.213 |
| Better support for splitting larg... | 20 | 245 | 76 | 0.310 | 245 | 23 | 0.094 |
| Run OSB on WebShere and ... | 21 | 237 | 36 | 0.152 | 237 | 142 | 0.599 |
| Browse and consume service... | 22 | 238 | 34 | 0.143 | 238 | 48 | 0.202 |
| Attach security policy for RES... | 23 | 237 | 45 | 0.190 | 237 | 37 | 0.156 |
| Be able to select multiple end... | 24 | 241 | 33 | 0.137 | 241 | 34 | 0.141 |

**Figure 10**: Max Diff Count Analysis

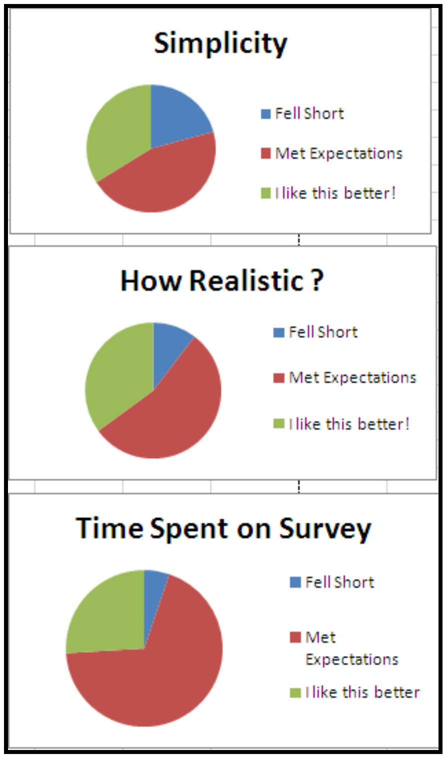



**Figure 11**: Result of MaxDiff Question about Survey

## Bibliography


Alba Joseph W, & Cooke Alan D J. (2004). When absence begets inference in conjoint analysis, *Journal of Marketing Research* Vol.XLI (November 2004), 382-387, XLI, 382-382-387.

Ayers, D. L. (2006). Conjoint analysis and its application in prioritizing software product features. Unpublished manuscript. (August 2006)

Bradlow, E. T. (2005). “Current issues and a 'wish list' for conjoint analysis,” *Appl. Stochastic Models Bus. Ind., 2005; 21:319-323, 21*, 319-323.

Carr, N. G. (2003). IT doesn't matter. *Harvard Business Review* (May): 41-49.

Cattin, P., Wittink, Richard. (1982). Commercial use of conjoint analysis. A survey. *Journal of Marketing 46*(Summer) 44-53.

Cattin, P., Wittink, Richard. (1989). Commercial use of conjoint analysis. An update. *Journal of Marketing 53*(3) 91-96.

Cohen, S. H. (2003). Maximum difference scaling: improved measures of importance and preference for segmentation. *Sawtooth Software Research Paper Series,* 1.

Dolan, Robert J. (1990). Conjoint analysis: a manager’s guide. *Harvard Business School,* HBS Case Services, No 9-590-059: 121-127.

Fortune. (2009). The Fortune 500's biggest winners. Last modified April 20, 2009. http://money.cnn.com/galleries/2009/fortune/0904/gallery.f500_mostprofitable.fortune/19.html.

Gownder, J. P. (2011). Introduction to conjoint analysis for consumer product strategists. Research Note Cambridge, MA: Forrester.

Green, P.E., Krieger AM, Wind. (2001) Thirty years of conjoint analysis: reflections and prospects. *Interfaces, 31*(May-June), S56-S73.

Green, P. E., Rao, V. R. (1971). Conjoint measurement for quantifying judgmental data. *Journal of Marketing Research Vol.VIII (August 1971)*, 355-63, VII, 355-63.

Green, P. E., Srinivasan, V. (1978). Conjoint analysis in marketing: New developments with implications for research and practice. *The Journal of Marketing, 54*(4), 3-19.

Hill, Aaron, Orme, B. K. (2011). *Choice modeling workshop*. Training Series, Sawtooth Software, Monterey, CA, July 2011.

Huber, Joel, (2005). Conjoint analysis: How we got here and where we are (an update). *Sawtooth Software.Technical Paper Series*. http://www.sawtoothsoftware.com/download/techpap/howwegot2.pdf

Johnson, Richard M., Orme, Brian (2007). A new approach to adaptive CBC. *Sawtooth Conference Proceedings*, pp 85-109. Sequim, Washington: Sawtooth Software.

Johnson Richard M. (August 2005). A career between theory and practice. *Journal of Marketing Research Vol.XLII (August 2005), 243-249, XLII*, 243-243-249.

Johnson Richard M. (May 1974). Trade-off analysis of consumer values. *Journal of Marketing Research Vol.XI (may 1974), 121-127, XI*, 121-121-127.

Meibner, M., Decker, R. (2009). An empirical comparison of CBC and AHP for measuring consumer preferences. *Proceedings of the International Symposium of the Analytical Hierarchy Process,* 370-391.

Netzer, O., O. Toubia, E.T. Bradlow, E. Dahan, T. Evgeniou, F.M. Feinberg, E.M. Feit, S.K. Hui , J. Johnson, J. C. Liechty, J.B. Orlin, V.R. Rao (2007), Beyond Conjoint Analysis: Advances in Preference Measurement. *Marketing Letters, 19*(3-4), 337-354.

Orme, B. K. (2010). *Getting started with conjoint analysis: Strategies for product design and pricing research* (Second ed.). Madison, WI 53713-1420: Research Publishers LLC.

Rossi, P. E, Allenby, G. M. (2003). Bayesian Statistics and Marketing. *Marketing Science, 22*(3), pp. 304-328.

Sawtooth Software (2011). Choice Modeling Workshop. (Training Series, Monterey, CA, July 2011)

Sawtooth Software. (2008). The CBC System for Choice-Based Conjoint Analysis. http://www.sawtoothsoftware.com/download/techpap/cbctech.pdf

Sawtooth Software. (2007). The MaxDiff/Web system technical paper. http://www.sawtoothsoftware.com/download/techpap/maxdifftech.pdf

Sethuranman, R., Kerin, R. A., & Cron, W. L. (2005). A field study comparing online and offline data collection methods for identifying product attribute preferences using conjoint analysis. *Journal of Marketing Research Vol.58 (2005), 602-610,* (58), 602-610.

Thurstone, L.L. (1927) A law of comparative judgment. *Psychological Review* 34(1927): 278–286.

West Patricia M, Brockett Patrick L, Golden Linda L. (1997). A comparative analysis of neural networks and statistical methods for predicting consumer choice. *Marketing Science, Vol 16, no.4.* (1997), Pp.370-391., 16(4), 370-370-391.